REMARKS

In accordance with the foregoing, claims 1, 2, 3, 5, 11, 21 and 23 have been amended. Claims 1-3, 5-11 and 21-23 are pending and under consideration.

With regard to item 5 of the Office Action "how often" has been changed to -- how many times --.

With regard with item 6 of the Office Action, "when obtaining" has been changed to -- in the process of obtaining --.

Claims 1-3, 6-11 and 21-23 are rejected under 35 U.S.C. 103(a) as being obvious over Knuth, "The Art of Computer Programming, Third Edition, Volume 1 (1997) in view of U.S. Patent No. 5,201,000 to Matyas et al. and Schneier, "Applied Cryptography, Protocols, Algorithms and Source Code in C" Second Edition, (1996) and Flanagan, "Java in a Nutshell," Third Edition (1999). The Examiner relies upon additional references to reject dependent claim 5.

As will be discussed below, even though the Examiner has relied upon four different references just to reject the independent claims, the present invention does not result. That is, even assuming that the Examiner's analysis is perfectly correct, none of the references teach increasing a base value by a value determined by an index previously stored.

The main difference between document Matyas et al. and the present invention is the way in which the two prime numbers are initially created, which is outlined in new claim 21 and 23, and how the two prime numbers are re-generated for producing a re-generated private key as outlined in claims 1 and 11.

In item 16 of the Office Action, the Examiner asserts that claims 1, 11, and 23 are substantially equivalent to claim 21. Regarding claim 21, the decisive steps are the step of checking, the step of repeating and, particularly, the step of storing an index. Regarding the functionality behind claim 21 for initially creating the public/private key pair, please refer to the paragraph bridging pages 10 and 11of the April 15, 2005 amendment. Regarding the re-generation (claims 1 and 11), please refer to the second paragraph on page 11 of the April 15, 2005 amendment.

As outlined in the first paragraph of page 11, there is a fundamental difference between the functionality of claim 21 for initially generating the keys and claim 1 for re-generating the keys. While claim 21 describes an iterative process, where the index is determined and stored during prime number generation, claim 1 describes a completely non-iterative process, in which the stored index is retrieved and used for increasing the base value for obtaining the first and second prime numbers. No iteration is performed in claim 1. Thus, the index is generated in claim 21 and used in claim 1. Therefore, the Examiner's statement under section 16 of the Office Action, that claim 1 is substantially equivalent to claim 21 is simply wrong.

To better understand why one would not modify Matyas et al. by storing an index indicating the number steps, one should understand in detail how the two prime numbers are generated in Matyas et al. Pertinent is the so-called passphrase mode, which is called mode "PP° under item 43 in Fig. 13.

The detailed way of generating the prime numbers is shown in Fig. 14 of Matyas et al. First of all, the user enters his password or passphrase (column 13, lines 61 and 62). Then, the input parameters are parsed, and a value CW from the input passphrase PP is generated. Then this value CW is sent to the key generation algorithm as outlined in column 19, lines 6 to 14 and shown in Fig. 14 under item 53. The key generation algorithm receiving value CW is shown in Fig. 7. The key generation algorithm receives value CW and checks whether this value is a prime number. This check is shown in step 162 of Fig. 7. CW is not a prime number, the control proceeds to "no" at step 162 in Fig. 7, and a new trial value is generated. To generate the new trial value, the key generation algorithm uses value CW to send a seed to the dynamically seeded pseudorandom number generator 200. Based on this seed, the dynamically seeded pseudorandom random number generator 200 outputs a value 56 and inputs this value into the key generation algorithm as a new trial value of P as shown in step 161 of Fig. 7. When this value is a prime number as determined by step 162 in Fig. 7, then control proceeds to "yes" at step 162. When, however, this value is not a prime number, control proceeds to "no" in step 162 and the key generation algorithm provides this "next seed" value received from item 200 as a new seed for the pseudorandom number generator 200. Based on this new seed, the pseudorandom number generator generates a new output at line 56, and this new output serves as a new trial value of P.

To summarize, a number of trials are performed until the prime number is found, which number of trials depends on the original value CW and, of course, the construction of the dynamically seeded pseudorandom number generator. It is important to note that no further information for re-generation is required in Matyas et al. Instead, the same key generation algorithm is performed in all devices as outlined in column 6, lines 19 to 21.

Referring to column 8, lines 31 to 33 of Matyas et al. states "in the present invention, the public and private keys are generated entirely from a passphrase that the user must remember."

Furthermore, column 8, lines 43 to 46 of the reference state, "the present invention makes possible the transportation of the keys from one device to another independent of any additional information other than the passphrase." As outlined in column 8, lines 46 to 48 of Maytas et al., the reference describes the ability to function without anything stored (such as an index) as a clear advantage of the Maytas et al. invention in contrast to the prior art of Maytas et al.

Contrary to Matyas et al. there are the following key features of the present invention:

Firstly, the algorithm for originally generating the public/private key pair is an iterative process for producing the index, while the method for re-generating the keys is a straight-forward non-iterative process using the stored index. Thus, in clear contrast to Maytas et al., the present invention requires different algorithms for generating the keys based on a passphrase.

Furthermore, the present invention generates, when originally calculating the key pair, the index and stores the index as outlined in the fifth step of claim 21. When re-generating the keys, in addition to the initial value, this stored index is required as clearly defined in claim 9, third step, where it is outlined that the base value is increased by a value determined by the index previously stored.

Thus, the present invention is different from Matyas et al. in the key features of Matyas et al. While document D1 requires the same algorithms on each device, the present invention requires different algorithms for originally generating and re-generating the keys. Furthermore, it is the key feature of Matyas et al. that the only information required is the passphrase so that a general portability is possible as outlined in column 8, lines 46 to 48. Contrary thereto, for re-generating the keys, the stored index is required in addition to the passphrase.

Therefore, document D1 cannot render obvious the present invention, since the inventive method/device is different in the key features of Matyas et al. Stated in other words, Matyas et al. teaches away from the present invention.

With regard to Knuth, the Examiner is referred to the section titled "algorithm P," which states "this algorithm has two distinct parts: Steps P1 to P8 prepare an internal table of 500 primes, and steps P9 to P11 print the answer in the form shown above."

Thus, Knuth only teaches to find the first 500 prime numbers and to print them as a table.

It would be complete nonsense to store the number of steps required for generating any prime numbers, since the number of steps is completely irrelevant in Knuth. Furthermore, Knuth indicates a deterministic process, which always starts at the first prime number as shown at step P1. Thus, one can perform the method of Matyas et al., as one likes. The number of steps to generate the table or the same numbers as the table will always be the same. Therefore, it would be completely useless and nonsense to store an index indicating how often the base value has been increased in Knuth.

Contrary thereto, in the claimed process, it is not guaranteed and will not happen that the method always starts at the same prime number, since the base value is not fixed to, for example, a value of "2" as in Knuth. With the claimed method, the initial prime number can vary depending on the predetermined initial value input by a user, (receiving step in claim 21) and the way of processing as defined in the processing step of claim 21.

Furthermore, the Examiner's attention is called to the fact that, when the Knuth method is started with any other prime number than the first prime number "2", then this process will not work, since, in step P6, previously determined prime numbers are required as the "possible prime divisors". This also becomes clear from the definition of the vector PRIME[K], which is set in step P1 and step P2, where, each time a prime is found, step P7 goes back to step P2.

Stated in other words, the Knuth process does not work at all, when one starts with any other value than the first prime number "2". This, however, will necessarily be the case, when claim 21 is reasonably construed, since the initial value entered by a user may be secret and, of course, should vary from user to user. i.e., be user-specific.

Furthermore, the step of processing as defined in claim 21 would not make any sense if the result would always be the same first prime number.

Furthermore, if the base value is always the first prime number, then the steps of repeating and storing would not make any sense, since the branching "when the base value is not a prime number" would never occur.

Therefore, Knuth does not have any relevance to the present invention.

Under section 11 of the Office Action, the Examiner states that those having ordinary skill in the art would have been motivated to combine Matyas et al. and Knuth, i.e., to replace the Fig. 14 key generation algorithm of Matyas et al. by Knuth. The Examiner states that this would be done "to generate a public/private key pair using a seed known to a user." This statement is technically wrong. Since Knuth is a completely deterministic process, it would contradict the central teaching of Matyas et al. Referring to column 6, lines 11 to 14 of Matyas et al., the reference states "During key generation, the passphrase is used as a seed value to generate the necessary random numbers used by the key generation algorithm in the process of generating the keys. No other random numbers are used by the key generation algorithm except those which are generated from the passphrase."

Matyas et al. wants random numbers. No random numbers occur in Knuth, Therefore, Knuth is not combinable to document D1 and vice versa,

Furthermore, in Matyas et al. the seed that is known to a user is different for each user. Since the passphrase is the secret in document D1, the Knuth algorithm would not work as outlined above. When, for example, a passphrase of a certain user results in any other number than 1, i.e., the first prime number, the process would not work, since the earlier-calculated prime numbers are required for doing the primality test in Knuth. When, for example, the Knuth method would have to decide the next prime number above the number "12", the algorithm in Knuth would require all prime numbers 2, 3, 5, 7 calculated beforehand which is clear from steps P5 and 6.

Therefore, using the Knuth method instead of the dynamically seeded pseudorandom number generator in Matyas et al. would not result in a public/private key pair, and therefore, the Examiner's conclusion under section 11 is not justified, since those skilled in the art would not combine anything to arrive at a completely useless solution.

Regarding section 14 of the Office Action, the Examiner is correct. However, the conclusion under section 15 is again not justified. In Flanagan, it is outlined that one loops through a data array. When one finds what one is looking for, the method remembers where it was found and the search was stopped.

The Examiner generalizes Flanagan stating in that "a number is stored representing how many times it has to check whether a certain condition is met". However, Flanagan is based on a data array, and the data array is searched for a certain data value. Then, the index of the data value data[i] is stored, when data[i] is equal to the target. Thus, i is an index indicating how many times the loop was executed (searched). Thus, it does not appear that the Examiner's generalization is correct.

To the contrary, the inventive index indicates, how often a base value has been increased until the first prime number or the second prime number is obtained. Flanagan is completely silent on any "increase" of a base value. Furthermore, Flanagan searches for a certain data word. Contrary thereto, in accordance with the present invention, the step of checking is a checking whether the base value is a prime number, etc. as defined in claim 21. Therefore, the Examiner's statement, i.e., the generalization of searching a data array and, storing an index is

PAGE 10/13 * RCVD AT 9/15/2005 10:47:04 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-6/0 * DNIS:8729306 * CSID:+ * DURATION (mm-ss):03-34

not justified.

Even if the Examiner's generalization were correct, it would not make any sense to combine Flanagan to Knuth for the following reasons.

As outlined above, the Knuth algorithm is a completely deterministic algorithm. When one starts with the first prime number "2", one will need always the same steps, until the 500 prime numbers or any other prime number is found. Therefore, it would not make any sense to count the number of steps.

Furthermore, when one starts at any number, which is not the first prime number, the Knuth process does not work at all, since all lower prime numbers are required for checking any numbers as outlined above. Finally, Knuth is not a prime number test algorithm determining whether an unknown base value is a prime number or not. Instead, Knuth is an algorithm for calculating and printing the first 500 prime numbers. Therefore, contrary to the Examiner's assertion in items 15, the complexity of an implemented program will not be decreased or computing efficiency increased. Instead, a completely useless method would be obtained.

Furthermore, one would not count the number of trials in Matyas et al., i.e., the number of seeding actions until the dynamically seeded pseudorandom number generator output is a prime number, and one would not store this count. This is because these actions are completely against the philosophy of Maytas et al., since Maytas et al. states that the password is the only information required for performing the process as outlined at column 8, line 32. Thus, it would be in contradiction to the wording of document D1 to count the number of trials and to store the count.

Furthermore, regenerating the keys using the count in Matyas et al. would not allow any direct calculation, since it is the very feature of the dynamically seeded pseudorandom number generator to calculate a new output based on the preceding seed. There is no way to calculate the result of, for example, four trials without explicitly performing these four trials.

Contrary thereto, the present invention as defined in claim 1 can directly calculate the prime numbers based on the base value and the value determined by the index previously stored and the predetermined increment.

Therefore, using the teachings of Flanagan in Matyas et al. is against the teachings of Matyas et al. and would result in a useless device.

All the above arguments also apply to claim 1. Furthermore, claim 1 has the limitation of the increasing step, which, as outlined in the claim, directly results in the first and second prime numbers. Nothing like that is mentioned in the prior art or has even been stated by the Examiner. Furthermore, no prior art discloses the use of the previously stored index of claim 1.

In view of all of the above reasons, it is submitted that the prior art rejections should be withdrawn. There being no further outstanding objections or rejections, it is submitted that the application is in condition for allowance. An early action to that effect is courteously solicted.

If there are any formal matters remaining after this response, the Examiner is requested to telephone the undersigned to attend to these matters.

If there are any additional fees associated with filing of this Amendment, please charge the same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: Sept 15 2005

By:
Mark J. Henry
Registration No. 36,162

1201 New York Ave, N.W., Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

CERTIFICATE OF FACSIMILE TRANSMISSION

I hereby certify that this correspondence is being transmitted via facsimile to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450
on Sept 15, 2005
STAAS & HALSEY
By:
Date : 9.15.05